JOURNAL OF COMPUTER AND SYSTEM SCIENCES 7, 481–496 (1973)

The Constructibility of a Configuration in a Cellular Automaton

TAKEO YAKU

Department of Mathematics, School of Science and Engineering, Waseda University, Shinjuku, Tokyo, Japan

Received June 21, 1972

A configuration is said to be with finite support if the states of all but finitely many cells in the array are quiescent. The results are as follows. It is recursively unsolvable when $d \geqslant 2$, for a configuration c with finite support in a d-dimensional cellular automaton, whether or not:

1. c is in the image of the parallel map for the cellular automaton.
2. c is in the image of the parallel map for the cellular automaton restricted to the set of configurations with finite support.

Further properties of parallel maps are also considered.

1. INTRODUCTION

A cellular automaton—also known as a tessellation atuomaton (Moore (1962), Myhill (1963), Yamada–Amoroso (1970, 1971), Amoroso–Cooper (1971, 1972), Ostrand (1971))—is a formalization of the concept of an infinite array of finite state machines, connected in a highly regular fashion with its neighbours. Each machine can synchronously change its state as a function of the states of the machines from which it can directly receive information. The simultaneous action of these "local" functions will define "global" functions which will act on the entire array, changing "patterns" of machine states in the array to other patterns. These state patterns are called "configurations."

The cellular automaton was originally employed by von Neumann in 1959 for study of self-reproduction. In [1], Moore presented the problem of machine self-reproduction in an abstract form with his concept of "self-reproducing configurations."

There are configurations which cannot occur except at time $T = 0$ under certain assumptions about the local function. That is, these configurations are not only unstable, but they are "nonconstructible" in the sense that there is no configuration at time $T - 1$ which is changed to the given configuration at time T by the global function. These configurations were introduced and named as "Garden of Eden" configurations by Moore. In other words, a configuration is Garden of Eden if it is in the complement of the image of the global function.

It is usually assumed that one state of a machine is "quiescent" in the sense that if all the states of a given machine and the machines directly connected with the machine are quiescent at time $T - 1$, then the state of the machine is quiescent at time T. A configuration is "with finite support" (also referred to as "finite") provided that the states of all but finitely many machines in the array are quiescent.

In [1, 2, and 3], Moore, Myhill and Amoroso–Cooper–Patt showed that the surjectivity of a global function is equivalent to the injectivity of a global function restricted to the set of configurations with finite support. In [4], however, Amoroso–Cooper proved that with respect to the set of configurations with finite support, the injectivity of a global function does not imply the surjectivity.

In [5], Amoroso–Patt gave the decision procedures for the surjectivity and the injectivity of global functions for one-dimensional cellular automata, i.e., one-dimensional arrays of finite state machines.

In [6], Kobuchi–Nishio proved that the image of a global function restricted to finite arrays, and some other sets of configurations are regular in a one-dimensional cellular automaton.

This paper deals with a few decidability questions of these sets. The results are as follows. It is recursively unsolvable when $d \geqslant 2$, for a configuration c with finite support in a d-dimensional cellular automaton, whether or not:

1. c is Garden of Eden, i.e., c is not in the image of the global function.

2. c is Garden of Eden with respect to the set of configurations with finite support, i.e., c is not in the image of the global function restricted to the set of configurations with finite support.

We also considered further properties of the set of configurations.

2. Preliminaries

For any set K and any integer i, K^i denotes the set of i-tuples of elements of K, and by Z we will denote the set of integers. When f is a mapping of a set K to a set L, $f|_{K'}$ denotes the restriction of f to a subset K' of K. For sets K and L, $K \times L$ denotes the Cartesian product of K and L, i.e., the set of two tuples (k, l) so that $k \in K$ and $l \in L$. We relate the definitions of cellular automata and Turing machines according to Yamada–Amoroso [7, 8], Codd [9] and Herman [10].

Definition 1. A *cellular automaton* is $A = (V, Z^d, X, f, q_0)$, where

(i) V is a finite nonempty set called the *state alphabet* of A. V represents the set of states that can be assumed by any machines being modeled.

(ii) d is a positive integer called the *dimension* of A. The elements of Z^d (i.e., d-tuples of integers), called *cells*, are used as names for the machines in the array.

(iii) X is a p-tuple of distinct d-tuples of integers and is called the *neighbourhood index*. It is used to define the uniform interconnection pattern among the machines in the array.

(iv) f is a mapping of a subset of V^p to V and is called the *local transformation* of A, satisfying $f(q_0, ..., q_0) = q_0$. Since the mapping f is considered as the subset $\hat{f} = \{(v_0, ..., v_{p-1}, v_p); (v_0, ..., v_{p-1}) \in V^p$ and $f(v_0, ..., v_{p-1}) = v_p\}$ of V^{p+1}, called the *derived set* of f, we will often define the mapping f by denoting its derived set $\hat{f}$ and an element of $\hat{f}$ is called a *production* of f.

(v) q_0 is an element of V called the *quiescent state* of A.

A cellular automaton A will be referred to as *d-dimensional* if d is the dimension of A.

A recursive function $c: Z^d \to V$ is called a *configuration* in A. The image $c(i)$ of $i \in Z^d$ will be referred to as the *state* of the cell i in the configuration c. A configuration c is *with finite support* provided that $\{i \in Z^d; c(i) \neq q_0\}$ is finite.

Configurations c_1 and c_2 will be called *shift-equivalent* if there exists a $k \in Z^d$ such that for any $i \in Z^d$, $c_1(i + k) = c_2(i)$ [7]. The equivalence classes of the set of configurations determined by the relation of shift-equivalence will be called *patterns*. The symbol [c] will denote the pattern containing the configuration c. It will convenient to represent (one-dimensional) patterns containing configurations with finite support as follows: If a configuration c with finite support is such that $c(k + i) = b_i$, $1 \leqslant i \leqslant j$, for some $k \in Z$ and $j > 0$, and if $c(k + i) = q_0$ for all $i < 1$ and all $i > j$, then $[c]$ can be represented by

$$\overline{q_0}b_1 \cdots b_j\overline{q_0}.$$

The mapping $N_X : Z^d \to (Z^d)^p$ defined as follows: If $X = (x_0, ..., x_{p-1})$ and $i \in Z^d$, then $N_X(i) = (i + x_0, ..., i + x_{p-1})$, where $i + x_k$ $(0 \leqslant k \leqslant p - 1)$ is the component wise sum of the d-tuples i and x_k, is used to specify the neighbourhood of any cell i relative to the neighbourhood index. For a configuration c, let c^p: $(Z^d)^p \to V^p$ be defined by

$$c^p(i_0, ..., i_{p-1}) = (c(i_0), ..., c(i_{p-1})).$$

Let C_A be the set of all configurations in A. The mapping $S_A : C_A \to C_A$ defined now from the local transformation f will be called the *parallel map* for A. For any configuration c, $S_A(c) = c'$ if and only if

$$c': Z^d \xrightarrow{N_X} (Z^d)^p \xrightarrow{c^p} V^p \xrightarrow{f} V.$$

Alternatively, for any $i \in Z^d$,

$$c'(i) = f(c^p(N_X(i))).$$

The mapping $(S_A)^n$ is defined recursively as follows:

(i) $(S_A)^1(c) = S_A(c)$,
(ii) $(S_A)^n(c) = S_A((S_A)^{n-1}(c))$ $(n > 1)$, where c is a configuration.

DEFINITION 2. A *Turing machine* is $T = (S, B, M, m_0, F, \alpha, \beta, \gamma)$, where

(i) S is a nonempty finite set called the *alphabet* of T.
(ii) $B \in S$ is the *blank symbol.*
(iii) M is a nonempty finite set called the *state set* of T.
(iv) $m_0 \in M$ is the *initial state* of T.
(v) F is a subset of M called the *terminal states* of T.
(vi) α, β, γ are functions of two variables, such that for any nonterminal state m and any $s \in S$,

$$\begin{array}{ll} \alpha(m, s) \in S & (\textit{new symbol function}) \\ \beta(m, s) \in \{-1, 1\} & (\textit{move function}) \\ \gamma(m, s) \in M & (\textit{next state function}). \end{array}$$

A *description* D of a Turing machine T is an expression of the form

$$m;\ s_{-u} \cdots \underline{\mathbf{s_w}} \cdots s_v\,,$$

where m is a state of T and $s_i \in S$, for $-u \leqslant i \leqslant v$ (u and v are nonnegative integers, $w \in Z$ and $-u \leqslant w \leqslant v$). The **bold** symbol is the one scanned by T. The description D is said to be terminal if m is terminal. If D is nonterminal, the consecutive description $S_T(D)$ is defined to be

$$\begin{array}{llll} \gamma(m, s_w);\ s_{-u} \cdots \underline{\mathbf{s_{w-1}}}\alpha(m, s_w)\, s_{w+1} \cdots s_v & \text{if}\ \ \beta(m, s_w) = -1 & \text{and} & -u < w \\ \gamma(m, s_w);\ \underline{\mathbf{B}}\alpha(m, s_w)\, s_{w+1} \cdots s_v & \text{if}\ \ \beta(m, s_w) = -1 & \text{and} & -u = w \\ \gamma(m, s_w);\ s_{-u} \cdots s_{w-1}\alpha(m, s_w)\, \underline{\mathbf{s_{w+1}}} \cdots s_v & \text{if}\ \ \beta(m, s_w) = 1 & \text{and} & v > w \\ \gamma(m, s_w);\ s_{-u} \cdots s_{w-1}\alpha(m, s_w)\underline{\mathbf{B}} & \text{if}\ \ \beta(m, s_w) = 1 & \text{and} & v = w. \end{array}$$

T is said to halt on $s_0 \cdots s_v$ ($v \geqslant 0$, $s_i \in S$ and $0 \leqslant i \leqslant v$), if there exists a sequence $D_0, \ldots, D_n$ of descriptions such that $D_0 = m_0\,;\ s_0 \cdots s_v\,,\ D_j = S_T(D_{j-1})$ $(0 < j \leqslant n)$ and D_n is terminal. The *blank tape halting problem* for a Turing machine is: Does the Turing machine halt on B? B is the *blank tape.*

3. Garden of Eden Configurations

Definition 3 (Amoroso–Cooper [4]). For a cellular automaton A, a configuration c is *Garden of Eden* if for any configuration d

$$S_A(d) \neq c, \qquad \text{where} \quad S_A \text{ is the parallel map for } A.$$

Theorem 1. *For any $d \geqslant 2$, there is no algorithm which, for any d-dimensional cellular automaton and any configuration in it with finite support, will decide whether or not the configuration is Garden of Eden.*

The proof consists of two parts. First we will construct the one-dimensional cellular automaton called the first simulator for a Turing machine T that simulates T in popular way [10, 11]. Next we will construct the two-dimensional cellular automaton from the first simulator for T, and show that the blank tape halting problem is equivalent to the problem whether or not a certain configuration in the two-dimensional cellular automaton is Garden of Eden.

Let $T = (S, B, M, m_0, F, \alpha, \beta, \gamma)$ be a Turing machine, and consider the one-dimensional cellular automaton $A_1 = (V_1, Z, X_1, f_1, q_0)$, called the *first simulator* for T, where

(i) $V_1 = (S \times (M \cup \{*\})) \cup \{\sigma, \tau, \#\}, \qquad q_0 = (B, *)$,

(ii) $X_1 = (-1, 0, 1)$,

(iii) f_1 is defined by Table I:

We cause A_1 to simulate T by embedding a configuration in it which "looks like" a description of T (see, e.g., [10, 11]). Thus, the embedded configuration c of a description $D = m; s_{-u} \cdots \underline{s_w} \cdots s_v$ is

$$[c] = \overline{q_0}\sigma(s_{-u}, *) \cdots (s_{w-1}, *)(s_w, m)(s_{w+1}, *) \cdots (s_v, *)\, \tau\overline{q_0}.$$

Consider the configuration $c_{\text{initial},1}$ such that

$$\begin{aligned} c_{\text{initial},1}(i) &= \# \qquad (i = 0) \\ &= q_0 \qquad \text{otherwise.} \end{aligned} \tag{1}$$

The first simulator A_1 for T is said to *terminate* if there exist integers $n > 0$ and i such that $(S_{A_1})^n(c_{\text{initial},1})(i) \in S \times F$. Since $[S_{A_1}(c_{\text{initial},1})] = \overline{q_0}\sigma(B, m_0)\,\tau\overline{q_0}$ simulates the description m_0 ; B of T ,we have the following lemma.

Lemma 1 (Herman [10], Smith III [11]). *The first simulator for a Turing machine terminates if and only if the given Turing machine halts on the blank tape.*

TABLE I

In the Table, $v, v', v_0, v_1, v_2 \in V_1$; $s_0, s_1, s_2 \in S$; and $m \in M$

v_0	v_1	v_2	$f_1(v_0, v_1, v_2)$	Conditions
$(s_0, *)$	$(s_1, *)$	$(s_2, *)$	$(s_1, *)$	
$(s_0, *)$	$(s_1, *)$	(s_2, m)	$\begin{cases}(s_1, *) \\ (s_1, \gamma(m, s_2))\end{cases}$	if $\beta(m, s_2) = -1$ if $\beta(m, s_2) = 1$ and $\gamma(m, s_2) \notin F$
$(s_0, *)$	(s_1, m)	$(s_2, *)$	$(\alpha(m, s_1), *)$	
(s_0, m)	$(s_1, *)$	$(s_2, *)$	$\begin{cases}(s_1, \gamma(m, s_0)) \\ (s_1, *)\end{cases}$	if $\beta(m, s_0) = -1$ and $\gamma(m, s_0) \notin F$ if $\beta(m, s_0) = 1$
q_0	q_0	$\#$	σ	
q_0	$\#$	q_0	(B, m_0)	
$\#$	q_0	q_0	τ	
q_0	q_0	σ	σ	
q_0	σ	v	$f_1(q_0, q_0, v)$	if $f_1(q_0, q_0, v)$ is defined
σ	v	v'	$f_1(q_0, v, v')$	if $f_1(q_0, v, v')$ is defined
v	v'	τ	$f_1(v, v', q_0)$	if $f_1(v, v', q_0)$ is defined
v'	τ	q_0	$f_1(v', q_0, q_0)$	if $f_1(v', q_0, q_0)$ is defined
τ	q_0	q_0	τ	

Otherwise f_1 is undefined.

The *second simulator* for a Turing machine $T = (S, B, M, m_0, F, \alpha, \beta, \gamma)$ is the two-dimensional cellular automaton $A_2 = (V_1, Z^2, X_2, f_2, q_0)$, where

(i) $X_2 = ((0, 0), (-1, 1), (0, 0), (1, 1))$

(ii) $$\begin{aligned} f_2(v_0, v_1, v_2, v_3) &= \# \quad \text{if } v_0 = \# \\ &= q_0 \quad \text{if } v_0 \neq \#, f_1(v_1, v_2, v_3) \text{ is defined and} \\ &\qquad\quad f_1(v_1, v_2, v_3) = v_0 \\ &= q \quad \text{otherwise,} \end{aligned} \tag{2}$$

where $v_0, v_1, v_2, v_3 \in V_1$, f_1 is the local transformation of the first simulator for T and q is an arbitrary nonquiescent state satisfying $q \neq \#$.

We will construct a configuration $c_{\text{index},2}$ in A_2 and prove that $c_{\text{index},2}$ is Garden of Eden if and only if A_1 terminates. The configuration $c_{\text{index},2}$ is defined to be of the form illustrated as Fig. 1.

$$
\begin{array}{ccccc}
 & \cdot & \cdot & \cdot & \\
\cdot & q_0 & q_0 & q_0 & \cdot \\
\cdot & q_0 & \# & q_0 & \cdot \\
\cdot & q_0 & q_0 & q_0 & \cdot \\
 & \cdot & \cdot & \cdot &
\end{array}
$$

FIGURE 1

Precisely,

$$
\begin{aligned}
c_{\text{index},2}(i,j) &= \# \qquad \text{if} \quad (i,j) = (0,0) \\
&= q_0 \qquad \text{otherwise.}
\end{aligned} \tag{3}
$$

LEMMA 2. *Let us suppose* $c_{\text{initial},1}$ *and* $c_{\text{index},2}$ *are the configurations defined in* (1) *and* (3), *respectively. If* $c_{\text{index},2}$ *is not Garden of Eden and* $S_{A_2}(d) = c_{\text{index},2}$ *for some configuration* d, *where* S_{A_2} *is the parallel map for* A_2, *then the following equation holds*:

$$
\begin{aligned}
d(i,j) &= \# \quad (\text{case } A\text{: } (i,j) = (0,0)) \\
&= \sigma \quad (\text{case } B\text{: } i < 0 \text{ and } j = i) \\
&= \tau \quad (\text{case } C\text{: } i > 0 \text{ and } j = -i) \\
&= (S_{A})^{-j}(c_{\text{initial},1})(i) \quad (\text{case } D\text{: } j < 0 \text{ and } j < i < -j).
\end{aligned}
$$

Proof. Cases A, B, and C are obvious by Table I. Now, we will prove case D by induction on j. This is obvious for $j = -1$. For any $j \geqslant -k$, suppose that the equation holds. First we will prove it for $i = -k$. From (3), $c_{\text{index},2}(-k, -k-1) = q_0$, accordingly, $d(-k, -k-1) = f_1(d(-k-1, -k), d(-k, -k), d(-k+1, -k))$. From case A, $d(-k, -k) = \sigma$, and from Table I, $d(-k-1, -k) = q_0$. While, $(S_{A_1})^k(c_{\text{initial},1})(-k-1) = (S_{A_1})^k(c_{\text{initial},1})(k) = q_0$. And therefore, by inductive hypothesis $d(-k, -k-1) = (S_{A_1})^{k+1}(c_{\text{initial},1})(-k)$. For each $i(-k < i \leqslant k)$, we can similarly prove that $d(-k-1, i) = (S_{A_1})^{k+1}(c_{\text{initial},1})(i)$. Q.E.D.

Proof of Theorem 1. Assume that $c_{\text{index},2}$ is not Garden of Eden. From Lemma 2, there exists the state $d(i,j)$ of each cell (i,j) for any $j < 0$ and any i $(j < i < -j)$, satisfying $d(i,j) = (S_{A_1})^{-j}(c_{\text{initial},1})(i)$.

Hence T does not halt on the blank tape, because if not and $(S_T)^n(m_0\,;\,B)$ is terminal for some $n > 0$, then there exists an integer i such that $(S_{A_1})^{n+1}(c_{\text{initial},1})(i)$is undefined. Accordingly, d is not defined for $(i, -(n+1))$, and d is not a configuration, since for any $v \in V_1$ $f_2(v, (S_{A_1})^n(c_{\text{initial},1})(i-1), (S_{A_1})^n(c_{\text{initial},1})(i), (S_{A_1})^n(c_{\text{initial},1})(i+1)) \neq c_{\text{index},2}(i, -(n+1))\,(=q_0)$, from Table I and (2). This contradicts the hypothesis, since d is arbitrary.

Conversely, if T does not halt on the blank tape, then the configuration e in A_2 satisfies $S_{A_2}(e) = c_{\text{index},2}$, where

$$\begin{aligned} e(i,j) &= \# && ((i,j) = (0,0)) \\ &= (S_{A_1})^{-j}\,(c_{\text{initial},1})(i) && (j < 0) \\ &= q_0 \quad \text{otherwise.} \end{aligned}$$

The theorem can be proved for $d = 2$ from the unsolvability of the blank tape halting problem (see, e.g., Minsky [12], p. 150).

Since a d-dimensional cellular automaton can be embedded to some $(d+1)$-dimensional cellular automaton, we can verify the theorem. Q.E.D.

4. Garden of Eden Configurations for Configurations with Finite Support

DEFINITION 4 (Amoroso–Cooper [4]). For a cellular automaton A, a configuration c in A is *Garden of Eden for configurations with finite support* if c is with finite support and for any configuration d in A with finite support,

$$S_A(d) \neq c, \qquad \text{where} \quad S_A \text{ is the parallel map for } A.$$

THEOREM 2. *For any $d \geqslant 2$, there is no algorithm which, for any d-dimensional cellular automaton and any configuration in it will decide whether or not the configurations is Garden of Eden for configurations with finite support.*

To prove this, we will construct the one-dimensional and the two-dimensional cellular automata, analogously to the proof of Theorem 1. And then a configuration with finite support corresponding to the configuration $c_{\text{index},2}$, defined by (3), will be found out. For a Turing machine $T = (S, B, M, m_0, \alpha, \beta, \gamma)$, the *third simulator* for T is the one-dimensional cellular automaton $A_3 = (V_3, Z, X_1, f_3, q_0)$ defined as follows:

(i) $V_3 = ((\{\sigma, \tau\} \cup S) \times (M \cup \{m_L, m_R, *\})$,

(ii) f_3 is defined by Table II. The productions of f_3 are classified to seven disjoint subsets; An element of each subset will be referred to as a production *of type n* $(n = 0, 1, \ldots, 6)$.

Consider the configuration $c_{\text{initial},3}$ in A_3 which is the same mapping as $c_{\text{initial},1}$ defined by (1). Precisely,

$$[c_{\text{initial},3}] = \overline{q_0}\#\overline{q_0}\,.$$

The third simulator A_3 for a Turing machine T is said to *terminate* if there is an integer $n > 0$ such that $(S_{A_3})^n(c_{\text{initial},3})(i) = q_0$ for all $i \in Z$.

TABLE II

In the Table, v_0, v_1, $v_2 \in V_3$; s_0, s_1, $s_2 \in S$; s_0', s_1', $s_2' \in S \cup \{\sigma, \tau\}$; $s_\sigma \in S \cup \{\sigma\}$; $s_\tau \in S \cup \{\tau\}$; and $m \in M$

Type	v_0	v_1	v_2	$f_3(v_0, v_1, v_2)$	Conditions
0	q_0	q_0	$\#$	$(\sigma, *)$	
	q_0	$\#$	q_0	(B, m_0)	
	$\#$	q_0	q_0	$(\tau, *)$	
1	$(s_\sigma, *)$	$(s_0', *)$	$(s_\tau, *)$	$(s_0', *)$	
	q_0	q_0	$(\sigma, *)$	q_0	
	$(\tau, *)$	q_0	q_0	q_0	
2	$(s_0', *)$	$(s_1', *)$	(s_0, m)	$(s_1', *)$	if $\beta(m, s_0) = 1$
				$(s_1', \gamma(m, s_0))$	if $\beta(m, s_0) = -1$ and $\gamma(m, s_0) \notin F$
				(s_1', m_L)	if $\beta(m, s_0) = -1$ and $\gamma(m, s_0) \in F$
	$(s_0',)$	(s_0, m)	$(s_1', *)$	$(\alpha(m, s_0), *)$	
	(s_0, m)	$(s_0', *)$	$(s_1', *)$	$(s_0', \gamma(m, s_0))$	if $\beta(m, s_0) = 1$ and $\gamma(m, s_0) \notin F$
				(s_0', m_L)	if $\beta(m, s_0) = 1$ and $\gamma(m, s_0) \in F$
				$(s_0', *)$	if $\beta(m, s_0) = -1$
	q_0	q_0	(σ, m)	q_0	if $\beta(m, B) = 1$ and $\alpha(m, B) = B$
				$(\sigma, *)$	if $\beta(m, B) = 1$ and $\alpha(m, B) \neq B$
				$(\sigma, \gamma(m, B))$	if $\beta(m, B) = -1$ and $\gamma(m, B) \notin F$
				(σ, m_L)	if $\beta(m, B) = -1$ and $\gamma(m, B) \in F$
	q_0	(σ, m)	$(s_0, *)$	$(\sigma, *)$	if $\beta(m, B) = 1$ and $\alpha(m, B) = B$
				$(\alpha(m, B), *)$	if $\beta(m, B) = 1$ and $\alpha(m, B) \neq B$
				$(\alpha(m, B), *)$	if $\beta(m, B) = -1$
	(σ, m)	$(s_0, *)$	$(s_0', *)$	$(s_0, \gamma(m, B))$	if $\beta(m, B) = 1$ and $\gamma(m, B) \notin F$
				(s_0, m_L)	if $\beta(m, B) = 1$ and $\gamma(m, B) \in F$
				$(s_0, *)$	if $\beta(m, B) = -1$
	$(s_0', *)$	$(s_1, *)$	(τ, m)	$(s_1, *)$	if $\beta(m, B) = 1$
				$(s_1, \gamma(m, B))$	if $\beta(m, B) = -1$ and $\gamma(m, B) \notin F$
				(s_1, m_L)	if $\beta(m, B) = -1$ and $\gamma(m, B) \in F$
	$(s_0, *)$	(τ, m)	q_0	$(\alpha(m, B), *)$	if $\beta(m, B) = 1$
				$(\tau, *)$	if $\beta(m, B) = -1$ and $\alpha(m, B) = B$
				$(\alpha(m, B), *)$	if $\beta(m, B) = -1$ and $\alpha(m, B) \neq B$
	(τ, m)	q_0	q_0	$(\tau, \gamma(m, B)$	if $\beta(m, B) = 1$ and $\gamma(m, B) \notin F$
				(τ, m_L)	if $\beta(m, B) = 1$ and $\gamma(m, B) \in F$
				q_0	if $\beta(m, B) = -1$ and $\alpha(m, B) = B$
				$(\tau, *)$	if $\beta(m, B) = -1$ and $\alpha(m, B) \neq B$

Table continued

TABLE II (*continued*)

Type	v_0	v_1	v_2	$f_3(v_0, v_1, v_2)$	Conditions
3	$(s_0', *)$	$(s_1', *)$	(s_τ, m_L)	(s_1', m_L)	
	$(s_0', *)$	(s_τ, m_L)	$(s_1', *)$	$(s_\tau, *)$	
	(s_τ, m_L)	$(s_0', *)$	$(s_1', *)$	$(s_0', *)$	
4	q_0	q_0	(σ, m_L)	q_0	
	q_0	(σ, m_L)	$(s_0, *)$	(σ, m_R)	
	(σ, m_L)	$(s_0, *)$	$(s_\tau, *)$	$(s_0, *)$	
5	q_0	q_0	(s_σ, m_R)	q_0	
	q_0	(s_σ, m_R)	$(s_0, *)$	q_0	
	(s_σ, m_R)	$(s_0, *)$	$(s_\tau, *)$	(s_0, m_R)	
6	q_0	q_0	(τ, m_R)	q_0	
	q_0	(τ, m_R)	q_0	q_0	
	(τ, m_R)	q_0	q_0	q_0	
Otherwise f_3 is undefined.					

LEMMA 3. *For a Turing machine $T = (S, B, M, m_0, F, \alpha, \beta, \gamma)$, the third simulator A_3 for T terminates if and only if T halts on the blank tape.*

Proof. Assume that T halts on the blank tape, and $D_0, \ldots, D_n$ $(n \geqslant 0)$ is the sequence of descriptions of T such that $D_0 = m_0; B$, $D_i = S_T(D_{i-1})$ $(0 < i \leqslant n)$, and D_n is terminal. Let us suppose that $c_{-1}, c_0, \ldots, c_n, \ldots$ is the sequence of configurations in A_3 such that $c_{-1} = c_{\text{initial},3}$, and $c_i = S_{A_3}(c_{i-1})$ $(i > -1)$.

Let π be the mapping of the set of descriptions of T to the set of configurations in A_3 defined as follows. The configuration $\pi(D)$ for a description $D = m; s_{-u} \cdots \underline{s_w} \cdots s_v$ $(u \geqslant 0, v \geqslant 0$ and $-u \leqslant w \leqslant v)$ is of the form

$$[\pi(D)] = \overline{q_0}(\sigma, x_{-u'-1})(s_{-u'}, x_{-u'}) \cdots (s_{v'}, x_{v'})(\tau, x_{v'+1})\overline{q_0},$$

where

$$\begin{aligned} u' &= u && \text{if } u = 0, \text{ or } u > 0 \text{ and } s_{-u} \neq B \\ &= u - 1 && \text{if } u > 0 \text{ and } s_{-u} = B, \\ v' &= v && \text{if } v = 0, \text{ or } v > 0 \text{ and } s_v \neq B \\ &= v - 1 && \text{if } v > 0 \text{ and } s_v = B, \\ x_w &= m, && \\ x_i &= * && (i \neq w \text{ and } -u' - 1 \leqslant w \leqslant v' + 1). \end{aligned}$$

Applying productions of types 1 and 2 in f_3, we have

$$S_{A_3}(\pi(D_i)) = \pi(S_T(D_i)), \qquad \text{for any} \quad i(0 \leqslant i \leqslant n-2).$$

Applying productions of types 0 and 1 in f_3, $S_{A_3}(c_{-1}) = \pi(D_0)$, and therefore

$$\pi(D_j) = c_j \qquad \text{for any } j \quad (0 \leqslant j \leqslant n-1).$$

If $D_n = m;\ s_{-u} \cdots \underline{s_w} \cdots s_v$ ($u \geqslant 0$, $v \geqslant 0$, $v \geqslant 0$ and $-u \leqslant w \leqslant v$), then c_n is the same as $\pi(D_n)$ except for $x_w = m_L$, i.e., $[c_n] = \overline{q_0}(\sigma, x_{-u'-1}) \cdots (\tau, x_{v'+1})\ \overline{q_0}$, where

$$\begin{aligned}
u' &= u && \text{if either } u > 0 \text{ and } s_{-u} \neq B, \quad \text{or} \quad u = 0\\
&= u - 1 && \text{if } u > 0 \text{ and } s_{-u} = B,\\
v' &= v && \text{if } v > 0 \text{ and } s_v \neq B \quad \text{or} \quad v = 0,\\
&= v - 1 && \text{if } v > 0 \text{ and } s_v = B,\\
x_w &= m_L\,, &&\\
x_i &= * && \text{if } i \neq w \text{ and } -u' - 1 \leqslant i \leqslant v' + 1.
\end{aligned}$$

In the transitions of configurations starting from c_n by consecutive applications of S_{A_3}, m_L goes left and reaches to the cell which has σ, from productions of type 3. Thus, we have

$$[c_{n+u'+w+1}] = \overline{q_0}(\sigma, m_L) \cdots (\tau, *)\overline{q_0}\,.$$

Applying the productions of type 4, we have

$$[c_{n+u'+w+2}] = \overline{q_0}(\sigma, m_R) \cdots (\tau, *)\overline{q_0}\,.$$

Next, applying the productions of type 5, m_R of $c_{n+u'+w+2}$ goes right consecutively and the cell where m_R exists changes its state to q_0. Every configuration in this phase is

$$[c_{n+u'+w+2+k}] = \overline{q_0}(s_{-u'+k}\,, m_R) \cdots (\tau, *)\overline{q_0}\,, \qquad \text{where} \quad 0 < k < u' + 1 + v'.$$

When m_R reaches to the cell which has τ, the cell changes its state to q_0, applying the productions of type 6. That is,

$$[c_{(n+u'+w+2)+(u'+1)+(v'+1)}] = \overline{q_0}(\tau, m_R)\overline{q_0}\,,$$

$$[S_{A_3}(c_{(n+u'+w+2)+(u'+1)+(v'+1)})] = \overline{q_0}q_0\overline{q_0}\,.$$

Hence A_3 terminates and the converse is similarly proved. Q.E.D.

For a Turing machine $T = (S, B, M, m_0, F, \alpha, \beta, \gamma)$, the *fourth simulator* for T is the two-dimensional cellular automaton $A_4 = (V_3, Z^2, X_2, f_4, q_0)$, defined by

$$\begin{aligned} f_4(v_0, v_1, v_2, v_3) &= \# \quad \text{if } v_0 = \# \\ &= q_0 \quad \text{if } v_0 \neq \#, f_3(v_1, v_2, v_3) \text{ is defined and } f_3(v_1, v_2, v_3) = v_0 \\ &= q \quad \text{otherwise,} \end{aligned} \tag{4}$$

where q is an arbitrary nonquiescent state satisfying $q \neq \#$, f_3 is the local map for the third simulator A_3, and $v_0, v_1, v_2, v_3 \in V_3$.

Proof of Theorem 2. Let us suppose $T = (S, B, M, m_0, F, \alpha, \beta, \gamma)$ is a Turing machine, and A_3 and A_4 are the third and the fourth simulators for T, respectively. Consider the configuration $c_{\text{index},4}$ in A_4 with finite support, defined by

$$\begin{aligned} c_{\text{index},4}(i, j) &= \# \quad \text{if } (i, j) = (0, 0) \\ &= q_0 \quad \text{if } (i, j) \neq (0, 0). \end{aligned}$$

We will show that $c_{\text{index},4}$ is Garden of Eden for configurations with finite support if and only if T does not halt on the blank tape.

Assume that $c_{\text{index},4}$ is such that $S_{A_4}(d) = c_{\text{index},4}$ for some configuration d with finite support, where S_{A_4} is the parallel map for A_4. Let d_j be the mapping of Z to V_3 defined by $d_j(i) = v$ iff $d(i, j) = v$ with respect to any $j \in Z$.

From Table II, if $f_3(v_0, v_1, v_2) \neq q_0$, then an element of $\{v_0, v_1, v_2\}$ is not quiescent, and therefore if $f_4(u, v_0, v_1, v_2) = q_0$ and $u \neq q_0$, then an element of $\{v_0, v_1, v_2\}$ is not quiescent from (4). Thus, if $d(k, j) \neq q_0$ for some integers k and $j > 0$, then an element of $\{d(k-1, j+1), d(k, j+1), d(k+1, j+1)\}$ is not quiescent since $c_{\text{index},4}(k, j)$ is quiescent. Accordingly, for all $n > j$, $d(m, n) \neq q_0$ for some $m \in Z$, and therefore d cannot be with finite support. Hence, $d(i, j) = q_0$ for all $i \in Z$ and $j > 0$.

Similarly, $d(i, 0) = q_0$ for all $i \neq 0$, and $d(0, 0) = \#$ from (4). Hence we have

$$d_0 = c_{\text{initial},3}.$$

Since $c_{\text{index},4}(i, 0) = q_0$ for all $i \neq 0$, then from (4), $d_{-1} = S_{A_3}(c_{\text{initial},3})$. Accordingly,

$$d_{-j-1} = S_{A_3}(d_{-j}), \qquad \text{for all} \quad j > 0.$$

Since d is with finite support then there is an integer $l > 0$ such that

$$d_{-l}(k) = q_0, \qquad \text{for all} \quad k \in Z.$$

Hence T halts on the blank tape, and the converse is similarly proved. Therefore, the theorem can be proved from the unsolvability of the blank tape halting problem. Q.E.D.

5. Erasable Configurations for Configurations with Finite Support

In [4], the notion of "erasable" is defined for the set of all configurations. Yet, we will define it only for the set of configurations with finite support.

Definition 5 (see Amoroso–Cooper [4]). A configuration c in a cellular automaton A is *erasable for configurations with finite support* if c is with finite support and there is a configuration c' with finite support such that

(i) $c' \neq c$

and

(ii) $S_A(c') = S_A(c)$,

where S_A is the parallel map for A.

Theorem 3. *For any $d \geqslant 2$, there is no algorithm which, for any d-dimensional cellular automaton and any configuration in it with finite support, will decide whether or not the configuration is erasable for configurations with finite support.*

This proof proceeds like that of Theorems 1 and 2. A configuration c is called *quiescent* if for any $i \in Z^d$, $c(i) = q_0$, where d is the dimension. We will construct the two-dimensional cellular automaton for a given Turing machine, and show that the quiescent configuration in it is erasable for configurations with finite support iff the Turing machine halts for the blank tape.

Let $A_3 = (V_3, Z, X_1, f_3, q_0)$ be the third simulator, defined in section 4, for a Turing machine $T = (S, B, M, m_0, F, \alpha, \beta, \gamma)$. The *fifth simulator* for T is the two-dimensional cellular automaton $A_5 = (V_3, Z^2, X_2, f_5, q_0)$ defined as follows:

$$\begin{aligned} f_5(v_0, v_1, v_2, v_3) &= q_0 \text{ if either } f_3(v_1, v_2, v_3) = v_0, \text{ or } (v_0, v_1, v_2, v_3) = (\#, q_0, q_0, q_0) \\ &= q \quad \text{otherwise,} \end{aligned} \tag{5}$$

where $v_0, v_1, v_2, v_3 \in V_3$, q is an arbitrary nonquiescent state, and f_3 is the local map for A_3. For the fifth simulator for T, we have the following lemma.

Lemma 4. *The quiescent configuration in the fifth simulator for a Turing machine T is erasable for configurations with finite support if and only if T halts on the blank tape.*

Proof. Assume that T halts on the blank tape. Then there is a sequence $c_0, c_1, \ldots, c_n$ $(n > 0)$ of configurations in the third simulator for T such that $c_0 = c_{\text{initial},3}$, $S_{A_3}(c_i) = c_{i+1}$ $(0 \leqslant i < n)$, and that c_n is quiescent.

Let c_q be the quiescent configuration in A_5 and d be the configuration in A_5 defined as follows:

$$\begin{aligned} d(i,j) &= q_0 && (j > 0, \text{ or } j < -n) \\ &= c_{\text{initial},3}(i) && (j = 0) \\ &= (S_{A_3})^{-j}\,(c_{\text{initial},3})(i) && (-n \leqslant j < 0). \end{aligned}$$

Clearly, $S_{A_5}(d) = c_q$. Note that $S_{A_5}(c_q) = c_q$.

Conversely, let c_q be erasable for configurations with finite support and c be a non-quiescent configuration with finite support such that $S_{A_5}(c) = c_q$. Since c is with finite support, then there exists an integer l such that $c(i,j) = q_0$ for any $i \in Z$ and $j > l$, and $c(i', l) \neq q_0$ for some $i' \in Z$. Similarly there is an integer k such that $c(i, l) = q_0$ for all $i < k$, and $c(k, l) \neq q_0$.

Since $c(k-1, l+1) = c(k, l+1) = c(k+1, l+1) = c_q(k, l) = q_0$, then $c(k, l) \neq \#$ by (5). From Table II, $c(k-1, l-1) = (\sigma, *)$. We can verify that $c(k+1, l) = q_0$, because if not and $c(k+1, l) = q$, then $f_3(q_0, \#, q) = c(k, l-1)$, but $f_3(q_0, \#, q)$ is not defined in Table II, a contradiction. Similarly, we can prove that $c(k+2, l) = q_0$.

From (5), $c(k, l-1) = (B, m_0)$ and $c(k+1, l-1) = (\tau, *)$. Let $j_{i,\sigma}$ and $j_{i,\tau}$ be integers defined by:

$$\begin{aligned} j_{i,\sigma} &= j + k \quad \text{if } (S_{A_3})^i\,(c_{\text{initial},3})(j) \in (\{\sigma\} \times (M \cup \{m_L, m_R, *\})) \cup ((S \cup \{\tau\}) \times \{m_R\}) \\ &= \text{undefined} \qquad \text{if} \quad (S_{A_3})^i\,(c_{\text{initial},3})(j) = q_0 \quad \text{for all } j \in Z, \end{aligned}$$

$$\begin{aligned} j_{i,\tau} &= j' + k \qquad \text{if} \quad (S_{A_3})^i\,(c_{\text{initial},3})(j') \in \{\tau\} \times (M \cup \{m_L, m_R, *\}) \\ &= \text{undefined} \qquad \text{if} \quad (S_{A_3})^i\,(c_{\text{initial},3})(j') = q_0 \quad \text{for all } j' \in Z. \end{aligned}$$

Now, for any $i > 0$, if $[(S_{A_3})^i(c_{\text{initial},3})] \neq \overline{q_0}q_0\overline{q_0}$, then $c(j_{i,\sigma} - 2, l - i) = c(j_{i,\sigma} - 1, l - i) = q_0$, since $c_q(j_{i,\sigma} - 1, l - i - 1) = q_0$, and neither $f_3(q, q', (\sigma, m))$ nor $f_3(q, q', (s, m_R))$ is defined for any q $(q \neq q_0)$, q' $(q' \neq q_0)$, $m \in M \cup \{m_L, m_R, *\}$ and $s \in S \cup \{\tau\}$, in Table II. Similarly, $c(j_{i,\tau} + 1, l - i) = c(j_{i,\tau} + 2, l - i) = q_0$ for any $i > 0$, if $[(S_{A_3})^i(c_{\text{initial},3})] \neq \overline{q_0}q_0\overline{q_0}$.

Thus, if $[(S_{A_3})^i(c_{\text{initial},3})] \neq \overline{q_0}q_0\overline{q_0}$ for $i > 0$, then we have

$$c(j, l - i) = (S_{A_3})^i\,(c_{\text{initial},3})(j), \qquad \text{for all} \quad j(j_{i,\sigma} \leqslant j \leqslant j_{i,\tau}).$$

Therefore T halts on the blank tape because if not, then $[(S_{A_3})^i(c_{\text{initial},3})] \neq \overline{q_0}q_0\overline{q_0}$ for any $i > 0$, accordingly for any $j > 0$ there exists $k \in Z$ such that $c(k, -j) \neq q_0$, which contradicts to the assumption that c is with finite support. Q.E.D.

Proof of Theorem 3. This is obvious by Lemma 4. Q.E.D.

DEFINITION 6. A configuration c in a cellular automaton A is *vanishing* if c is nonquiescent and with finite support, and there is a positive integer n such that $(S_A)^n(c)$ is quiescent, where S_A is the parallel map for A.

THEOREM 4. *For any $d \geqslant 2$, it is unsolvable if a configuration with finite support in in a d-dimensional cellular automaton is vanishing.*

Proof. This theorem follows to Lemma 3. Q.E.D.

There is a vanishing configuration if and only if the quiescent configuration is erasable for configurations with finite support. Then, from Lemma 4 we have

THEOREM 5. *It is unsolvable for any $d \geqslant 2$, if there is a vanishing configuration in a d-dimensional cellular automaton.*

6. CONCLUDING REMARKS

Finally we note the following. Let us suppose $C_{A,F}$ and C_A denote the set of configurations with finite support and the set of configurations in a cellular automaton A, respectively. Then we can restate the main results of this paper in the following way. For a d-dimensional cellular automaton A, it is recursively unsolvable when $d \geqslant 2$, if

(i) a configuration c with finite support is in the image $S_A(C_A)$ of the parallel map $S_A : C_A \to C_A$.

(ii) a configuration c with finite support is in the image $S_A(C_{A,F})$ of the parallel map $S_A : C_A \to C_A$ restricted to $C_{A,F}$.

In [5], Amoroso–Patt gave the decision procedures for the surjectivity and the injectivity of parallel maps for one-dimensional cellular automata, i.e., in the former, they showed that the emptiness problem for the image of a parallel map is solvable if the dimension is one.

ACKNOWLEDGMENT

The author wishes to express deepest appreciation to Professor H. Noguchi of Waseda University for his kind advice in the preparation of the manuscript. He is also indebted to Professor K. Kobayashi and Mr. T. Itoh in revising the paper.

REFERENCES

1. E. F. MOORE, Machine models of self-reproduction, *Proc. Symp. Appl. Math.* **14** (1962), 17–33.
2. J. MYHILL, The converse of Moore's Garden-of-Eden theorem, *Proc. Amer. Math. Soc.* **14** (1963), 685–686.

3. S. AMOROSO, G. COOPER, AND Y. N. PATT, Some comments on the concept of a Garden-of-Eden configuration, unpublished.
4. S. AMOROSO AND G. COOPER, Garden-of-Eden theorem for finite configurations, *Proc. Amer. Math. Soc.* **26** (1970), 158–164.
5. S. AMOROSO AND Y. N. PATT, Decision procedures for surjectivity and injectivity of parallel maps for tessellation structures, *J. Comput. System Sci.* **6** (1972), 448–464.
6. Y. KOBUCHI AND E. NISHIO, Properties of certain state sets in the system of one-dimensional iterative automata (in Japanese), *Electron. Commun. Japan* **54-c** (1971), 396–403 (revised in *J. Information Sci.* **5** (1973), 199–216).
7. H. YAMADA AND S. AMOROSO, A completeness problem for pattern generation in tessellation automata, *J. Comput. System Sci.* **4** (1970), 137–176.
8. H. YAMADA AND S. AMOROSO, Structural and behavioral equivalences of tessellation automata, *Information and Control* **18** (1971), 1–31.
9. E. F. CODD, "Cellular Automata," Academic Press, Inc., New York and London, 1968.
10. G. T. HERMAN, Computing ability of a developmental model for filamentous organisms, *J. Theoret. Biol.* **25** (1969), 421–435.
11. A. R. SMITH III, Simple Computation-Universal cellular spaces, *J. ACM* **18** (1971), 339–353.
12. M. L. MINSKY, "Computation: Finite and Infinite Machines," Prentice-Hall, Inc., Englewood Cliffs, N. J., 1967.
13. S. AMOROSO AND G. COOPER, Tessellation structures for reproduction of arbitrary patterns, *J. Comput. System Sci.* **5** (1971), 455–464.
14. T. J. OSTRAND, Pattern reproduction in tessellation automata of arbitrary dimension, *J. Comput. System Sci.* **5** (1971), 623–628.
15. J. VON NEUMANN, "Theory of self-reproducing automata," completed and edited by A. W. Burks, University of Illinois Press, Urbana, Ill., 1966.